

Medidor monofásico de energía

Ricardo Pacheco, *Estudiante Miembro IEEE*, y Héctor Franco, *Estudiante Miembro IEEE*

*Escuela de ingeniería Eléctrica, Universidad de El Salvador
San Salvador, El Salvador*

Pacheco9264@gmail.com
francokhfe@gmail.com

Resumen — Desarrollo de un prototipo medidor de energía monofásico doméstico, mediante una plataforma de desarrollo modular de hardware y software libre, El medidor de energía permite realizar mediciones de valores rms de tensión y corriente eléctrica, además de valores de potencia activa y aparente junto con la energía total consumida durante un periodo de tiempo determinado; el usuario es capaz de acceder a estos valores de manera inalámbrica a través de una interfaz tanto en tiempo real como a datos documentados en la memoria del medidor, estos datos almacenados son importados a un archivo reporte que contiene tabuladas las mediciones y la energía total consumida en W/h.

Temas claves—Desarrollo modular de hardware y software, Medición de energía, IC ADE7753, comunicación inalámbrica, módulos XBee, Estándar IEEE 802.15.4, interfaz gráfica, LabVIEW, Almacenamiento y manejo de datos, Arduino, eficiencia energética.

I. INTRODUCCIÓN

En la situación energética regional actual, en la cual el encarecimiento de la producción es constante adjunto al mal uso del recurso eléctrico, es necesario buscar prontas y útiles soluciones, pero la responsabilidad sobre soluciones a esta problemática no sólo depende de la diversificación de fuentes de energía sino también del uso responsable de esta, conocer el consumo real efectuado por una instalación, máquina, electrodoméstico o circuito eléctrico en general es un punto clave para diagnosticar problemas de eficiencia energética, en el hogar.

Desde esta perspectiva, el dispositivo está orientado a domótica, específicamente en el área del ahorro energético, por medio de la medición de consumo la cual permite realizar auditorías domésticas sencillas y prácticas.

El presente proyecto propone un medidor de consumo de energía eléctrica el cual fue desarrollado con fines didácticos enmarcado en el panorama expuesto anteriormente, con la finalidad de crear un medidor de bajo costo y de bajo consumo.

En el mercado se encuentran medidores que persiguen un fin parecido a este (medidores domésticos de bajo costo y bajo consumo), pero una de las principales características de este proyecto es la forma de desarrollo, la cual fue implementada haciendo uso de una plataforma de desarrollo modular, utilizando software y hardware libre Arduino, dotando al dispositivo de flexibilidad en su diseño permitiendo explorar diversas áreas de aplicación y tecnologías, manteniendo definitivamente bajos los costos tanto en el desarrollo de prototipos como en el de dispositivos terminados.

El prototipo desarrollado incluye cuatro módulos principales:

- Medición de energía monofásica.
- Sistema de documentación y almacenamiento de datos.
- Comunicación inalámbrica.
- Interfaz gráfica de usuario.

II. MÓDULOS PRINCIPALES

Para comprender mejor el funcionamiento del medidor se presenta una descripción de los principales tópicos que caracterizan al prototipo, profundizando en puntos clave que permiten el acoplamiento de hardware y software en el dispositivo integrando los diferentes módulos.

A. Medición de energía

Esta función es implementada por el circuito integrado ADE7753.[1].

Se utiliza la tarjeta Arduino para que funcione como dispositivo controlador o dispositivo Master del IC ADE7753, el cual se encarga de realizar los muestreos de la señal y de realizar los cálculos correspondientes para obtener parámetros de *voltaje*, *corriente*, *potencia* y *energía*, luego los guarda en los registros de memoria que posee y los actualiza continuamente; el trabajo de Arduino consiste entonces en poder realizar lecturas y escrituras de configuración y de registros de datos, todo esto se lleva a cabo bajo el estándar de comunicación SPI

que soportan ambos dispositivos.

El IDE Arduino ya incorpora la librería <SPI.h> que por ser de código abierto, la configuración necesaria para poder comunicarse con el IC se basa en 4 sentencias principales que dependen particularmente de los dispositivos que van a funcionar en modo esclavo.

En el micro controlador Arduino los pines son como sigue:

- *MOSI Master Out Slave In.* Es el pin D11.
- *MISO Master In Slave Out.* Es el pin D12.
- *SCLK Serial Clock.* Es el pin D13.
- *SS Slave Selector.* Puede ser cualquier pin digital conveniente.

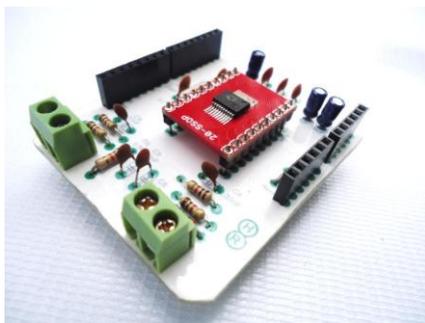


Figura 1. Shield Arduino medidor ADE7753

Para conocer los parámetros es necesario leer cuidadosamente las especificaciones de los fabricantes del IC; para el ADE7753 tenemos que la primera sentencia importante es:

```
SPI.setDataMode(SPI_MODE2);
```

En general, hay 4 modos de transmisión. Estos modos controlan si los datos salen o entran en un flanco de subida o bajada de la señal de reloj, esto se llama 'fase del reloj'. Y cuando el reloj está libre es decir su estado inicial se llama polaridad del reloj.

TABLA I
LOS CUATRO MODOS DIFERENTES DE TRABAJO
DEL RELOJ.

MODO	Polaridad del Reloj CPOL	Fase del Reloj CPHA
SPI_MODE0	0	0
SPI_MODE1	0	1
SPI_MODE2	1	0
SPI_MODE3	1	1

El ADE7753 utiliza el modo 2 por tanto si **CPHA = 0**, el reloj de cambio es la OR de SCLK con la terminal Slave Selector. Tan pronto como el terminal Slave Selector se coloca en un nivel lógico 0, los nuevos datos se ponen en la línea y el primer filo del reloj se leen los datos. Si **CPOL** se activa a un nivel lógico '1', el primer borde de reloj baja y los bits de datos subsecuentes se leen en cada filo de bajada sobre la línea de reloj. Cada nuevo bit se pone en la línea cuando el reloj tiene un flanco ascendente de Reloj.

La segunda sentencia importante es la configuración de la frecuencia del SCLK o los pulsos de reloj para la comunicación SPI.

```
SPI.setClockDivider(SPI_CLOCK_DIV32);
```

Esta sentencia configura el divisor del reloj SPI relativo al reloj del sistema, para Arduino la frecuencia del sistema es de 16MHz, por lo tanto la frecuencia correspondiente que se le ha configurado al SCLK es de 500KHz.

Otra sentencia que es clave para poder realizar la comunicación con el ADE7753 es el orden en que salen y entran los bits.

```
SPI.setBitOrder(MSBFIRST);
```

Esto es importante puesto que configura si los bits más significativos van primero o si lo menos significativos se envían primero, por supuesto tanto en la escritura como en la lectura se tiene la misma configuración.

La cuarta sentencia inicializa el bus SPI, declarando como salidas SCLK, MOSI, SS; poniendo SCLK y MOSI en bajo y en alto el SS. Pero para que la configuración del Slave Selector tenga lugar debe hacerse como se muestra en la segunda sentencia a continuación.

```
SPI.begin();
```

```
SPI.begin(slaveSelectPin)
```

Para este caso específico se usa la primera sentencia pues se tiene más de un dispositivo esclavo.

B. Almacenamiento de datos

Se requiere tener la capacidad de almacenamiento de datos para que estos puedan ser descargados en cualquier momento, para esto se hace uso de un Datalogger que básicamente se conforma de dos componentes básicos que son: un módulo de memoria SD (o Micro SD) y un módulo RTC (Real Time Clock) que servirá para poder guardar los datos con el tiempo en que fueron adquiridos.

El incluir estos módulos conlleva dos implicaciones principales, en primer lugar está el hecho que los pines para el almacenamiento en la memoria SD son los mismos que los utilizados por el ADE7753, es decir ambos dispositivos utilizan el protocolo SPI de comunicación, por ello es necesario utilizar los “selectores de chip” o “slave selector” de ambos dispositivos de manera que cada vez que el Arduino necesite comunicarse con alguno de ellos en específico los pines del otro dispositivo tienen que estar en alta impedancia o “eléctricamente desconectados” de esa manera se asegura que la comunicación se está realizando con un solo dispositivo a la vez.

Se utiliza el pin D9 del Arduino como SS del IC porque la shield de la Datalogger viene configurada para utilizar el pin D10. Por lo tanto el esquema ilustrativo físico de conexión es el que se muestra en la figura 2, cada vez que se necesite comunicarse con uno de los dispositivos se tiene que acompañar de los siguientes comandos:

```
digitalWrite(9,LOW);
```

“aquí instrucciones de comunicación con dispositivo.”

```
digitalWrite(9,HIGH);
```

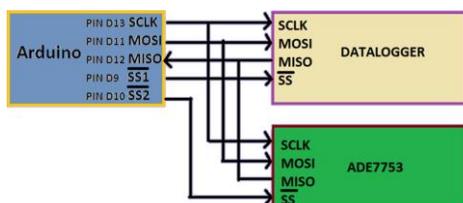


Figura 2. Esquema de conexión entre Arduino-Datalogger-ADE7753.

La segunda implicación aparece a la hora de integrar estos dispositivos, y está relacionado a las

librerías que utiliza la Datalogger, estas son <SD.h><Wire.h><RTClib.h> las dos últimas no generan problema alguno, pero en el código de la librería <SD.h> aparentemente no se define la polaridad del reloj de trabajo que tendrá la comunicación SPI por lo tanto significa que por defecto trabaja con “SPI_MODE0”, es decir el almacenamiento de datos en una SD se realiza con una polaridad distinta a la que utiliza el circuito integrado. Según la documentación encontrada sobre este tema este no suele ser un problema común debido a que todas las shields que se hacen para Arduino siempre trabajan con la misma polaridad de reloj, por tanto la peculiaridad del IC de trabajar con “SPI_MODE2” impide que la comunicación se lleve a cabo, siendo insuficiente los selectores de chip.

La solución para esto es la siguiente:

1- La inicialización de la comunicación SPI con el integrado debe preceder a la inicialización de la Datalogger.

Lo que esto significa es que el comando “while (!Serial)” y el comando “if (!SD.begin(10))” que sirven para inicializar la comunicación con la SD deben de estar después de haber hecho las declaraciones de polaridad del reloj para el integrado y del comando “SPI.begin()”

2- Cada vez que se guarden datos en la SD debe ir precedido del correspondiente comando de cambio de polaridad y al finalizar la comunicación se deberá restablecer la polaridad para que la comunicación con el IC sea posible.

```
SPI.setDataMode(SPI_MODE0);
```

“Aquí instrucciones para guardar o leer datos.”

```
SPI.setDataMode(SPI_MODE2);
```

C. Comunicación inalámbrica

Para esta característica se utilizan módulos XBee los cuales utilizan el conjunto de protocolos ZigBee basados en el estándar IEEE 802.15.4 de redes inalámbricas de área personal. Estos protocolos de alto nivel de comunicación inalámbrica, son utilizados para aplicaciones de radio difusión digital de bajo consumo.

Este protocolo proyecta sus principales

aplicaciones en domótica debido sus características:

- Bajo consumo.
- Topologías de redes malla.
- Fácil integración electrónica.



Figura 3. Módulo xbee serie 1

Se pueden construir nodos ZigBee con muy poca electrónica, los nodos ZigBee pueden desarrollar funciones de *coordinador*, *router* o *dispositivo final*; La función que desarrolla un nodo específico es configurada de la manera deseada en cada dispositivo; los nodos coordinadores se encargan de controlar la red y los caminos que deben seguir los dispositivos para comunicarse entre ellos, los routers interconectan a los dispositivos separados de la topología de la red y los dispositivos finales solo pueden comunicarse con su coordinador o su router, no pueden comunicarse directamente con otros dispositivos.

El bajo consumo de energía se debe a que los nodos ZigBee, en especial los configurados como dispositivos finales, pueden permanecer dormidos la mayor parte del tiempo, y cuando se requiere su participación estos pueden despertar en un tiempo muy corto aproximadamente de 15ms, para volver a dormir cuando haya realizado su función.

Algunas topologías ejemplo son:

- Topología estrella: el coordinador se coloca en el centro dejando dispositivos finales a su alrededor.
- Topología árbol: El coordinador es la raíz mientras que dispositivos routers interconectan las ramas.
- Topología malla: al menos uno de los nodos tiene más de dos conexiones.

Esta última es la que más ventajas ofrece debido a su versatilidad, si en un momento dado, un nodo falla, la comunicación puede seguir otros caminos utilizando los otros nodos interconectados; esta gestión de caminos es función de coordinador.

Para el medidor se utilizan:

- 2 dispositivos XBee (serie uno): estas son las antenas que transmiten y reciben frames de datos con protocolo ZigBee.

- 1 tarjeta XBee Shield: esta permite acoplar la antena XBee a la placa Arduino conectando los pines de comunicación serial de ambos dispositivos directamente.
- 1 XBee explorer Dongle: esta permite conectar directamente la antena XBee al puerto USB sin necesidad de cables.

Los dispositivos Xbee han sido configurados para trabajar de manera transparente, es decir que la comunicación es directa entre el puerto USB de la PC y el medidor Arduino, tal como si estos estuvieran cableados entre sí.

Si bien el prototipo desarrollado no explota las capacidades de los módulos Xbee por ser un medidor monofásico único (de un solo punto), los protocolos ZigBee abren una inmensa gama de posibilidades que se adecuan definitivamente a los fines buscados, posibilitando el desarrollo de mediciones multipunto así como ampliar la funcionalidad de los dispositivos más allá de la medición, pudiendo incluir funciones de control.

D. Plataforma de desarrollo modular

Como se ha mencionado antes, el desarrollo modular permite explorar diversas áreas de aplicación y tecnologías, manteniendo bajos los costos tanto en el desarrollo de prototipos como en el de dispositivos finales.

El prototipo desarrollado incluye cuatro módulos principales:

- Medición de energía monofásica.
- Sistema de documentación y almacenamiento de datos.
- Comunicación inalámbrica.
- Interfaz gráfica de usuario.

Cada módulo es independiente, y todos son integrados y gestionados desde Sketch (Código Arduino) montado el microcontrolador Arduino, el cual se encarga de comunicarse con cada módulo y coordinar el flujo de comunicación entre ellos.

El desarrollo de cada módulo goza de autonomía, es decir, que cada módulo puede ser cambiado por otro que cumpla las características de diseño deseadas, sin intervenir en el funcionamiento de los otros módulos. Esto permite dentro de las limitaciones de la tarjeta Arduino, la inclusión de diversos dispositivos que pueden ir desde circuitos integrados dedicados para una función específica, hasta la comunicación con una PC y otras máquinas. Logrando así dotar al dispositivo con características

variadas y flexibles a cambios de diseño, reuniendo en un prototipo específico características concretas que alcancen los fines del proyecto, y una vez alcanzados estos fines, los módulos pueden ser unificados en un mismo circuito como un diseño o producto final.

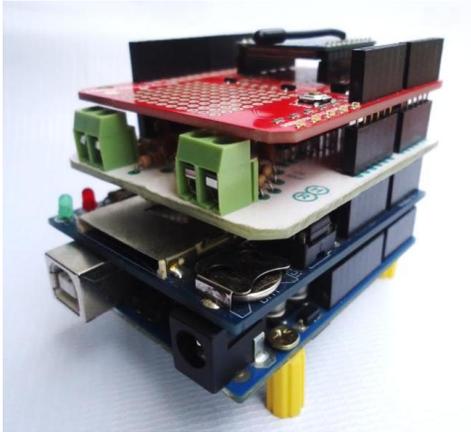


Figura 4. Plataforma de desarrollo de abajo hacia arriba: tarjeta Arduino UNO, Datalogger Shield, ADE7753 Shield, Xbee Shield.

En el desarrollo del proyecto se utilizó una tarjeta Arduino UNO como base de desarrollo, debido a que ha sido diseñada para este tipo de tratamiento de hardware, disponiendo de puertos analógicos y digitales, sobre los cuales se puede montar directamente otros módulos que son llamados “Shields”, Arduino UNO cuenta con las siguientes características de comunicación:

Utiliza el IC ATmega328 el cual ofrece comunicación serial UART TTL (5V), que está disponible en los pines digitales 0 (RX) y 1 (TX). Un ATMEGA16U2, esta comunicación serial a través de USB aparece como un puerto COM virtual en el ordenador. El firmware '16U2 utiliza los controladores de COM USB estándar, y no se necesita ningún controlador externo. Sin embargo, en Windows, se requiere un archivo. Inf. El software de Arduino incluye un monitor de puerto serie que permite a los datos de texto simples para ser enviados hacia y desde la placa Arduino. Los LEDs RX y TX de la placa parpadearán cuando los datos se transmiten a través del chip USB a serie y la conexión USB al ordenador (pero no para la comunicación en serie en los pines 0 y 1).

Una biblioteca SoftwareSerial permite la comunicación en serie en cualquiera de los pines digitales del UNO.

El ATmega328 también soporta la comunicación I2C (TWI) y SPI. El software de Arduino incluye librerías para simplificar el uso del bus I2C, y la comunicación SPI.

El prototipo desarrollado utiliza la comunicación serial UART en los pines digitales 0 y 1, para interactuar con la PC por medio de los dispositivos XBee. Además se utiliza comunicación SPI para los módulos de medición y de almacenamiento de datos, como se mencionó antes.

E. Interfaz de usuario

La interfaz de usuario desarrollada en LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) el cual es un entorno de desarrollo que ingenieros y científicos utilizan para programación gráfica y la integración de hardware, para diseñar y desplegar rápidamente sistemas de medidas y control.

Principales características de LabVIEW:

- Lenguaje desarrollado para medición, control y automatización
- Ambiente de desarrollo intuitivo para incrementar la productividad
- Fácil Integración con miles de instrumentos y dispositivos de medición
- Ambiente abierto para usar con otras aplicaciones
- Compilado para optimizar el desempeño del sistema

El Toolkit NI LabVIEW Interface para Arduino (LVIFA) ayuda a establecer una interfaz fácilmente con el microcontrolador Arduino usando LabVIEW.

Con este juego de herramientas y LabVIEW, se puede controlar y adquirir datos desde el microcontrolador Arduino. Una vez que la información está en LabVIEW, se puede analizar usando los cientos de bibliotecas integradas de LabVIEW, es posible desarrollar algoritmos para controlar el hardware Arduino y presentar conclusiones en un VI.

Un sketch para el microcontrolador Arduino actúa como un motor de E/S que se conecta con los VIs de LabVIEW a través de una conexión serial. Esto le ayuda a mover información rápidamente desde pines Arduino a LabVIEW sin ajustar la comunicación, la sincronización o incluso una sola línea de código C. Al usar Open, Read/Write, Close en LabVIEW, se puede tener acceso a las señales digitales, analógicas, moduladas por ancho de pulso, I²C y SPI del microcontrolador Arduino.

Este juego de herramientas no hace posible una operación autónoma, es decir que el microcontrolador Arduino debe estar conectado a la PC con LabVIEW a través de un enlace USB, serial, Bluetooth o XBee, esto implica una desventaja para el desarrollo del proyecto, ya que no permitiría al Arduino realizar mediciones independientemente ni documentarlas en la memoria. Por esta razón aun disponiendo de estas herramientas no se utilizan dentro del desarrollo del prototipo.

La comunicación entre LabVIEW y el medidor se logra mediante comunicación serial, utilizando las herramientas de comunicación con los puertos de la PC disponibles en LabVIEW, logrando una comunicación transparente entre la interfaz de usuario y el medidor.

La interfaz gráfica en LabVIEW es uno de los componentes primordiales del proyecto siendo el medio principal de comunicación e interacción entre el usuario y el dispositivo.

El software diseñado en LabVIEW posee habilidades de comunicación inalámbrica con el dispositivo medidor, a través del módulo XBee conectado al puerto USB por medio del XBee Explorer Dongle.

Gráficamente la interfaz ha sido diseñada para brindar una experiencia agradable, con menús y cuadros de texto fácilmente legibles, explotando las cualidades de los elementos de presentación que posibilitan utilizar una misma área para presentar diferentes ventanas haciendo uso de selectores y botones, permitiendo optimizar el espacio y dar a la vez una apariencia y flujo de cuadros interesante.

El menú principal utiliza iconos gráficos haciendo la selección más intuitiva, y la mayoría de ventanas se presentan como pestañas invisibles para mostrar y ocultar grupos de controles e indicadores lo que brinda una mayor estética.

La interfaz permite al usuario una serie de opciones para tener control sobre las mediciones que se efectúan en el dispositivo medidor.

Para permitir al usuario interactuar con el dispositivo y navegar dentro de las diferentes opciones la interfaz dispone de un menú gráfico en el cual se puede seleccionar de entre 4 opciones diferentes:



Figura 5. Etapa de detección del medidor.

1. **INICIO:** en esta ventana es posible iniciar la comunicación con el dispositivo por medio de una prueba de conectividad, que define si se está o no en el rango de alcance del medidor.

2. **MEDIR:** la segunda opción permite realizar mediciones en tiempo real, es decir el usuario puede visualizar las lecturas actuales en el momento que lo desee.

3. **DESCARGAR:** esta opción está diseñada para descargar el archivo en el cual se encuentran documentadas las lecturas realizadas desde la última vez que se borró el archivo.

4. **SALIR:** en esta ventana se muestran los principales datos y créditos de los desarrolladores, además de ser el menú de salida del sistema.



Figura 6. Etapa de medición en tiempo real.

III. PROTOTIPO Y RESULTADOS

El prototipo termina se muestra en la figura 7, aquí se incluyen las pinzas de voltaje y la pinza de

corriente, y se muestra a los módulos dentro de una carcasa de protección.

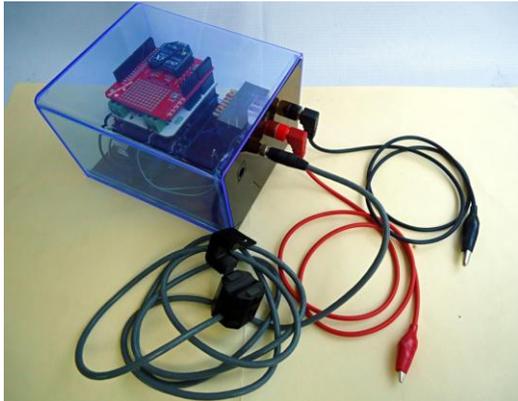


Figura 7. Prototipo medidor de consumo monofásico doméstico.

El medidor fue sometido a una prueba que consistió en censar el consumo de una carga de específica durante dos días a una frecuencia de documentación de un dato cada cinco segundos. Cabe señalar que este dato no es una lectura instantánea sino que es un valor resultante del proceso de cálculo realizado por el ADE7753.

TABLA II
RESUMEN DATOS TÉCNICOS DE LA CARGA DE PRUEBA.

Dato técnico	Valor
Consumo de entrada	135 W
Corriente total	2.2 A

En total se documentaron 34,323 datos capturados cada 5 segundos la tabla III presenta una muestra de los datos documentados por el medidor, durante los dos días de prueba, estos solo representan una pequeña parte de los 34,323 datos documentados.

TABLA III
MUESTRA DE DATOS COLECTADOS POR EL MEDIDOR.

Voltaje [V]	Corriente [A]	Potencia Activa [W]	Potencia Aparente [VA]	Fecha [dd/mm/hh]	Hora [hor:min:seg]
118.9	2	144.87	199.75	22/06/2013	20:13:40
118.92	2	144.22	199.75	22/06/2013	20:13:45
118.88	1.99	144.22	198.9	22/06/2013	20:13:50
118.86	1.99	144.22	198.9	22/06/2013	20:13:55
118.9	1.99	143.58	198.9	22/06/2013	20:14:00
118.9	1.99	143.58	198.9	22/06/2013	20:14:05

118.88	1.99	143.58	198.9	22/06/2013	20:14:10
118.85	1.99	142.94	198.9	22/06/2013	20:14:15
118.88	1.99	143.58	198.05	22/06/2013	20:14:20
118.88	1.99	142.94	198.05	22/06/2013	20:14:25

Del total de datos capturados se presentan en las imágenes 8 a 9 las gráficas de las lecturas obtenidas. Estas presentan el comportamiento de los valores de tensión, corriente, potencia activa y potencia aparente contra tiempo en horas.

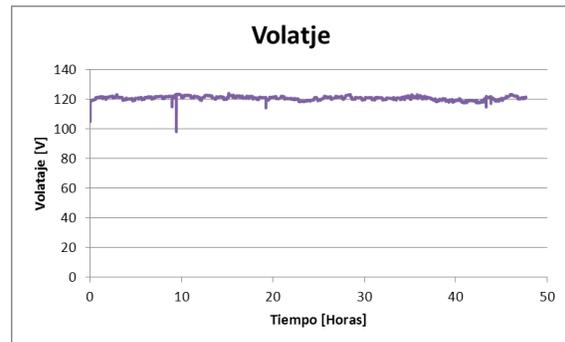


Figura 8. Gráfico de Voltaje vs tiempo.

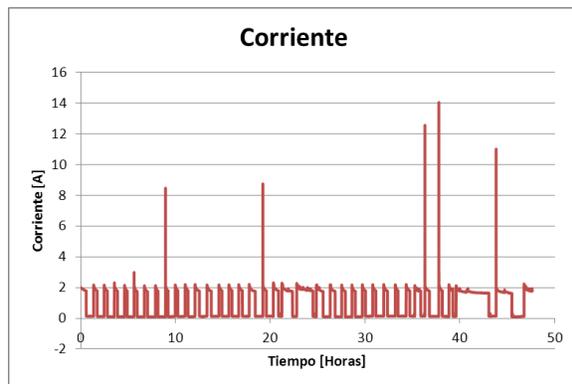


Figura 9. Gráfico de Corriente vs tiempo.

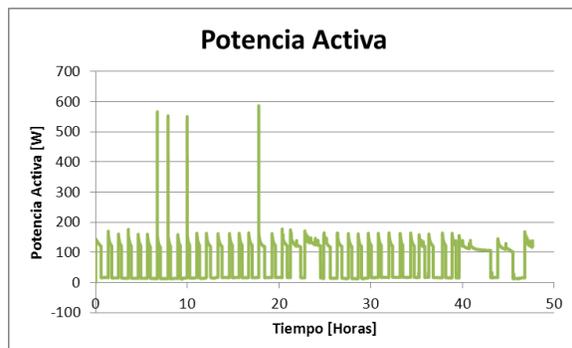


Figura 8. Gráfico de Potencia Activa vs tiempo.

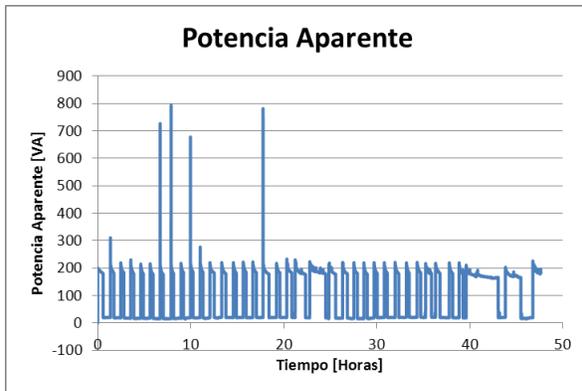


Figura 8. Gráfico de Potencia Aparente vs tiempo.

RECONOCIMIENTOS

Agradecemos a nuestro catedrático guía Ing. Wilber Calderón por su persistente apoyo, el cual dio como resultado la exitosa culminación de este proyecto.

A nuestras familias por el soporte económico y motivacional que sin dudar nos han brindado.

REFERENCIAS

- [1] Analog Devices, “Single-Phase Multifunction Metering IC with di/dt Sensor Interface”, ADE7753. One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A. ©2003–2010 Analog Devices, Inc. All rights reserved.

BIOGRAFÍAS



Héctor Franco, nació en San Salvador, departamento de El Salvador, el 29 de octubre de 1990. Se graduó del Colegio Salarrué en Sonsonate, El Salvador y estudió en la Universidad de El Salvador.

Su experiencia profesional incluye formar parte del equipo de desarrollo dentro del departamento de Informática y Documentación de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador, además de formar parte del equipo técnico de prácticas de laboratorio en el área de electrónica para la misma universidad. Sus áreas de interés incluyen, entre otras, fuentes de generación alternativas.



Ricardo Pacheco, nació en San Salvador, departamento de El Salvador, el 30 de marzo de 1992. Se graduó del C.E. INSA en Santa Ana, El Salvador y estudió en la Universidad de El Salvador.

Su experiencia profesional incluye formar parte del equipo técnico de prácticas de laboratorio en el área de electrónica para la Universidad de El Salvador, Sus áreas de interés incluyen, entre otras, energías renovables.